

## CLAIMS

500  
A

1. In a dynamic programming system, the improvement comprising:

a tree data structure implemented in a computer-readable memory  
5 accessible by a processor, said tree data structure having plurality of nodes that topologically define a root node and a plurality of parent-child generations, including a deepest child generation that is topologically furthest from the root; and

a traversal algorithm implemented by said processor, said  
10 algorithm traversing said nodes based on a set of traversal rules whereby nodes of a given generation are processed before the parent nodes of said given generation are processed, the deepest child generation is processed first, and traversal among nodes of each generation proceeds in the same topological direction.

2. The improvement of claim 1 wherein said tree data structure is encoded in said memory as a flat representation in which nodes of each generation occupy contiguous memory locations.

5

3. The improvement of claim 1 wherein said tree data structure is encoded in said memory with parent-child generations being represented through linked list.

10

4. The improvement of claim 1 wherein said tree data structure is encoded in said memory as a flat representation in which nodes of each generation occupy contiguous memory locations and have indicia designating the topological boundary between children of the same parent.

15

5. The improvement of claim 1 further comprising a mechanism for designating selected ones of said nodes as active nodes.

20

6. The improvement of claim 5 wherein said mechanism for designating selected ones of said nodes as active nodes comprises an active envelope data structure associated with said tree data structure.

7. The improvement of claim 1 further comprising a mechanism for designating selected ones of said nodes as active nodes and wherein said traversal algorithm includes a traversal rule whereby only active nodes are processed.

5

8. The improvement of claim 1 wherein said tree data structure is a lexical tree representing a lexicon.

9. The improvement of claim 1 wherein said tree data structure is a lexical tree representing the lexicon of a speech recognizer.

SA  
A1

10. In a dynamic programming system, the improvement comprising:

15 a tree data structure implemented in a computer-readable memory accessible by a processor, said tree data structure having plurality of nodes;

a mechanism for designating selected ones of said nodes as active nodes; and

20 a traversal algorithm implemented by said processor, said algorithm traversing said nodes based on a set of traversal rules whereby only said active nodes are traversed.

11. The improvement of claim 10 wherein said mechanism for designating selected ones of said nodes as active nodes comprises an active envelope data structure associated with said tree data structure.

5

12. The improvement of claim 10 wherein said traversal algorithm includes a dynamic programming process that assigns a likelihood score to nodes that are traversed.

10

13. The improvement of claim 12 wherein said mechanism for designating selected ones of said nodes uses said likelihood score to designate said active nodes.

15

14. The improvement of claim 10 wherein said traversal algorithm includes a dynamic programming process that assigns a likelihood score to nodes that are traversed and wherein nodes are designated as active nodes if their likelihood score is above a predetermined threshold.

20

15. The improvement of claim 14 wherein said predetermined threshold is calculated based on the highest likelihood score.

SB  
AI

16. The improvement of claim 10 wherein said a mechanism for designating selected ones of said nodes as active nodes defines an active envelope and uses a set of rules to propagate the active envelope by removing nodes that have a likelihood score below a predetermined  
5 threshold.

17. The improvement of claim 10 wherein said a mechanism for designating selected ones of said nodes as active nodes defines an active envelope and uses a set of rules to propagate the active envelope by  
10 inserting nodes that have a likelihood score above a predetermined threshold.

18. The improvement of claim 17 wherein said set of rules for inserting nodes guarantees that the nodes in said active envelope are  
15 sorted by their topological index.

SB  
AI

19. The improvement of claim 1 wherein said processor employs a cache and wherein said wherein said tree data structure is encoded in said memory such that traversal of said tree proceeds into  
20 said cache.

